

Timestamp Synchronization of Received Frames among Multiple Wireless LAN Nodes for Robust Access Point Coordination

Kazuto Yano[†]Eiji Nii[†]Kenta Suzuki[†]Koji Yamamoto[‡][†]Wave Engineering Laboratories, Advanced Telecommunications Research Institute International

2-2-2 Hikaridai, Seika, Souraku, Kyoto 619-0288, Japan

[‡]Graduate School of Informatics, Kyoto University

Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

{kzyano, e-nii, kenta-suzuki}@atr.jp, kyamamoto@i.kyoto-u.ac.jp

Abstract—Timestamp of frame reception timing of each wireless communication node is useful information to realize effective access control in a coordinated manner among multiple nodes. In order to obtain fine frame reception timestamp from multiple nodes with an independent clock, this paper proposed a method to compensate for the timestamp error due to clock drift among multiple nodes. The proposed method forms a node tree to determine the order of timestamp synchronization using the information of the commonly received frames among the nodes. From the root to the leaf node(s) in the formed tree, the timestamps are synchronized applying linear regression to the timestamps of commonly received frames for the parent and child nodes. The tree is formed so that the number of commonly received frames increases in order to improve the accuracy of the linear regression. Then, the jitter performance of the original timestamp obtained at each node is evaluated by experimental evaluation in an anechoic chamber using four different node configurations. This experimental evaluation revealed that timing synchronization function (TSF) timestamp is obtained with a jitter of a few μ s. Finally, the system-level simulation assuming IEEE 802.11n WLAN was conducted to evaluate the residual timestamp error among the coordinating nodes. The simulation results revealed that the proposed method can keep the residual timestamp error around 1 μ s if TSF stamp is available and the clock drift rate is stable in a couple of minutes.

Index Terms—IEEE 802.11, wireless LAN, access point coordination, TSF timestamp, timestamp synchronization

I. INTRODUCTION

IEEE 802.11 wireless LANs (WLANs) [1] has been widely and densely deployed, and its demand is still growing. The current WLAN generally employs a distributed and autonomous channel access mechanism based on carrier sense multiple access with collision avoidance (CSMA/CA). Therefore, increase of traffic demand in the WLAN brings severe contention to obtain a transmission opportunity among multiple basic service sets (BSSs) sharing the same radio channel, and it leads to frequent collision of frame transmission.

To improve the spectrum efficiency in dense overlapping BSS (OBSS) situations of the WLAN, a centralized coordination among multiple access points (APs) has attracted increased attention. For example, in the standardization of the next generation of the IEEE 802.11 WLAN (IEEE 802.11be),

several types of multi-AP coordination schemes including coordinated orthogonal frequency division multiple access (OFDMA), which share radio resources in a frequency domain by splitting into multiple portions and allocated to multiple BSSs [2], [3].

Moreover, several studies on AP coordination employing a network management entity which is placed beyond a backhaul and performs centralized control on multiple BSSs [4] were conducted. For example, the reference [5] proposed a scheme to control the threshold of clear channel assessment. The reference [6] proposed a scheme to allocate operating channels to multiple APs. Reinforcement learning (RL) is also employed in multi-AP coordination. The reference [7] proposed a method to allocate different channels to BSSs within the carrier sensing range of each AP to improve throughput.

However, there are many cases in which there is a node, i.e., AP or station (STA) out of the centralized control. If such uncoordinated node is operated around the coordinating APs, it is hard to make proper decision on BSS control or frame transmission. Hence, in order to overcome this problem, the reference [8] has proposed a new AP-coordination method. It introduces a recent robust reinforcement learning framework called robust adversarial reinforcement learning (RARL) [9]. In this method, a coordinator learns not only a transmission policy of the coordinating AP, that of the uncoordinated AP using RARL on the assumption that the uncoordinated AP tries to disturb the transmission of the coordinated AP by causing a frame collision.

In the above scheme, the coordinator needs to collect the information on the period of each frame transmission in the coordinating BSSs and its transmitter in order to know when and which node transmits a frame. However, there is clock drift between any two nodes; it leads to timestamp error between the two nodes and the error will become large time-by-time even if the clock between the two nodes is once synchronized by some protocol such as network time protocol (NTP). Moreover, a timestamp issued on an operating system (OS) fluctuates according to the instantaneous computational

load of the node. On the other hand, the timestamp error among the coordinating nodes needs to be μs -order in order to effectively manage the channel access among them because the frame duration of the current IEEE 802.11 WLAN ranges from tens to hundreds of μs .

To overcome this problem, this paper proposes a method to compensate for the timestamp error among the nodes in cooperation. In the proposed method, the coordinator collects some information about the received frames such as the received time and the information to identify each received frame. Then, the proposed method forms a node tree to determine the order of timestamp synchronization using the information of the commonly received frames among the nodes. Finally, the proposed method synchronizes the timestamps among the nodes sequentially from the root to the leaf node(s) according to the formed tree structure.

In order to validate the proposed method, this paper first conducts an experimental evaluation of jitter performance for two kinds of timestamps, i.e., epoch time issued on an OS and timing synchronization function (TSF) timestamp [1] that can be obtained on some wireless LAN devices. Then, the timestamp error performance of the proposed method is evaluated through assuming an overlapping BSS (OBSS) environment of IEEE 802.11 WLAN and taking account for the jitter performance obtained in the experimental evaluation.

The remainder of this paper is as follows. Section II introduces the proposed timestamp synchronization method. Section III explains the settings and results of the experimental evaluation of the jitter performance of epoch time and TSF timestamp. Section IV shows the result of the system-level simulation evaluating the timestamp error performance of the proposed method. Finally, Section V gives the conclusion of this paper.

II. PROPOSED TIMESTAMP SYNCHRONIZATION METHOD

In this paper, it is assumed that each node in cooperation sends its log of frame reception to the coordinator. The log of frame reception includes the reception time (timestamp) and some information that is useful to identify the unique frame such as transmitter ID, receiver ID, frame type, sequence number, and retry flag, of each received frame. Using the logs collected from the coordinating nodes, the coordinator counts up the number of frames that are received by both nodes in each pair of two coordinating nodes, and then it forms a tree for timestamp synchronization from the results of the counting up.

Let $\mathcal{V}$ denote the index set of nodes, and $\mathcal{E} = \{ \{m, n\} \mid m, n \in \mathcal{V} \text{ and } m \neq n \}$ denote the set of edges representing adjacent node pairs. Let the index set of frames transmitted in the system be denoted by $\mathcal{I}$, and the index set of frames received by node $n \in \mathcal{V}$ be denoted by $\mathcal{I}_n \subseteq \mathcal{I}$. At each frame reception, node n issues a timestamp $s_i^{(n)}$ for frame $i \in \mathcal{I}_n$. Note that even for the same frame i , timestamps issued by multiple nodes are not necessarily identical due to clock drift.

The purpose of the proposed timestamp synchronization is to compensate the drift. Here we utilize linear regression using

the timestamps of the commonly received frames $\mathcal{I}_m \cap \mathcal{I}_n$ for adjacent nodes $\{m, n\} \in \mathcal{E}$. The key idea here is that the accuracy of the linear regression increases as the number of commonly received frames increases.

Figure 1 shows an example of the synchronization tree. Each value in this figure indicates the number of commonly received frames in the corresponding node pair. The coordinator determines the direction and order of timestamp synchronization using this tree as follows.

Step-1: Find the adjacent node pair which has the largest number of commonly received frames,

$$\{m^*, n^*\} = \arg \max_{\{m, n\} \in \mathcal{E}} |\mathcal{I}_m \cap \mathcal{I}_n|, \quad (1)$$

where $|\cdot|$ represents the cardinality. Then, for each node of the selected node pair, the sum of the number of commonly received frames for other adjacent nodes is calculated as follows:

$$\sum_{\substack{\{m^*, l\} \in \mathcal{E}_m \\ l \neq n^*}} |\mathcal{I}_{m^*} \cap \mathcal{I}_l|, \quad \sum_{\substack{\{n^*, l\} \in \mathcal{E}_n \\ l \neq m^*}} |\mathcal{I}_{n^*} \cap \mathcal{I}_l|. \quad (2)$$

The node with the greater number becomes the root of the synchronization tree, and the other node becomes a child node of the root.

Step-2: Find the node which has the largest number of commonly received frames with any node already in the existing synchronization tree. The selected node joins the synchronization tree as the child the corresponding node which gives the largest number of commonly received frames. This step is repeated until any node cannot be added into the existing synchronization tree(s).

Step-3: Apply Step-1 and subsequently Step-2 to the nodes which are not included into any synchronization tree to create a new tree until no node can be added into the existing tree(s). (It should be noted that the control for the node which does not join to any tree is not necessary at the time because such node does not interfere to other coordinating node.)

After forming the synchronization tree(s), a synchronization process is performed from the root to the leaf node(s) for each synchronization tree. Since, the clock drift is stable for a couple of minutes in general, the synchronization process employs linear regression to derive the relationship between the timestamps of the frames obtained at the child node and that obtained at its parent node. By using the derived relationship, the timestamps of the frames received at the child node is converted onto the timeline on its parent node.

III. EXPERIMENTAL EVALUATION OF TIMESTAMP JITTER PERFORMANCE

As shown in Fig. 2, two kinds of timestamp are available as the timestamp of frame reception timing for IEEE 802.11 WLAN. One is “epoch time” that is issued as an OS-level

TABLE I
CONFIGURATION OF RECEIVING NODES

Node	PC	OS	WLAN device	Sniffer	TSF timestamp
Win 7 laptop	Fujitsu Lifebook E742/F	Windows 7	NETGEAR 6210-100JPS	Omnipeek v9.2	N/A
Win 10 laptop	Dell Vostro Notebook 3490	Windows 10	NETGEAR 6210-100JPS	Omnipeek v20.1	N/A
Mac laptop	Apple Macbook Pro 16-inch	macOS Catalina	AirMac Extreme	Wireless Diagnostics	Available
Linux laptop	Lenovo ThinkPad X1 Carbon Gen 8	Ubuntu 20.04.01 LTS	Intel AX201	Wireshark	Available

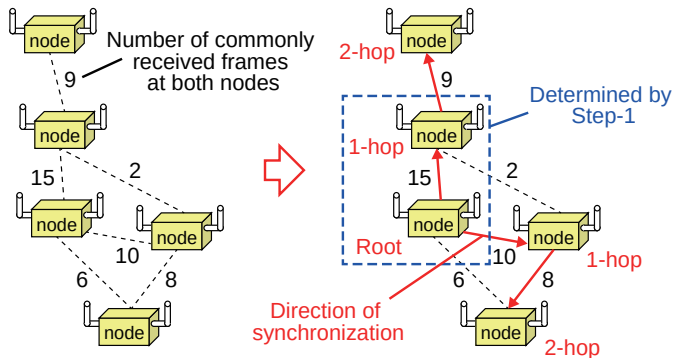


Fig. 1. Example of tree for timestamp synchronization.

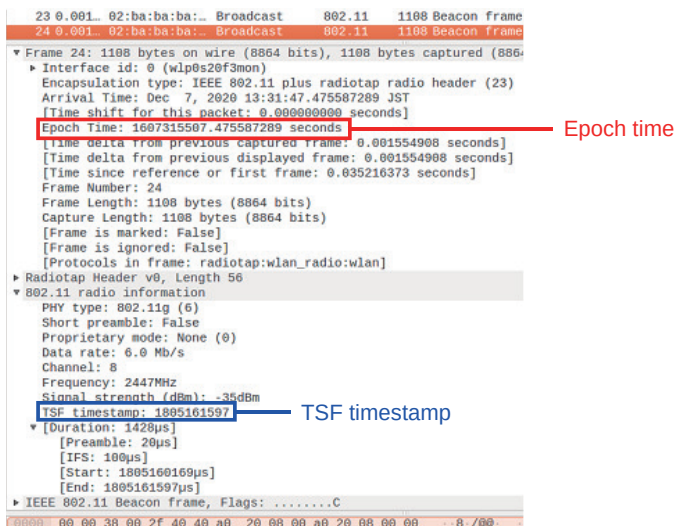


Fig. 2. Example of capture result of IEEE 802.11g frame.

timestamp. The other one is “TSF timestamp” [1] that is issued as a device-level timestamp. Epoch time is a universal timestamp, and it can be available at any node configuration such as OS, and WLAN device. However, it can fluctuate due to computational load of node and/or OS-level process interrupt. On the other hand, TSF timestamp is supported by only some WLAN devices, whereas it is expected to be stable because it is not affected by OS-level process interrupt.

Since the synchronization accuracy of a timestamp is highly affected by the jitter of the original timestamp, we conducted an experimental evaluation to measure the distribution of jitter at real nodes with different configurations. The experimental setup and a photo of the experimental site is shown in Figs. 3

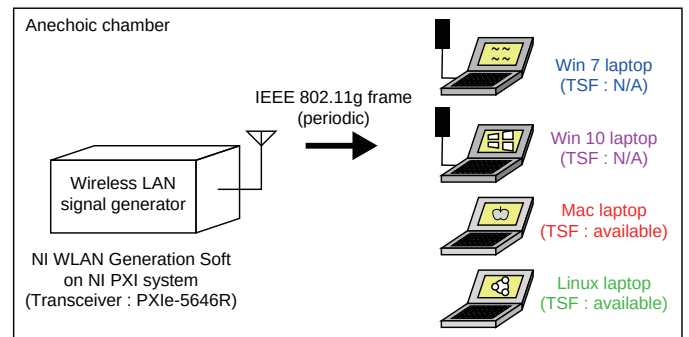


Fig. 3. Setup of experiment.

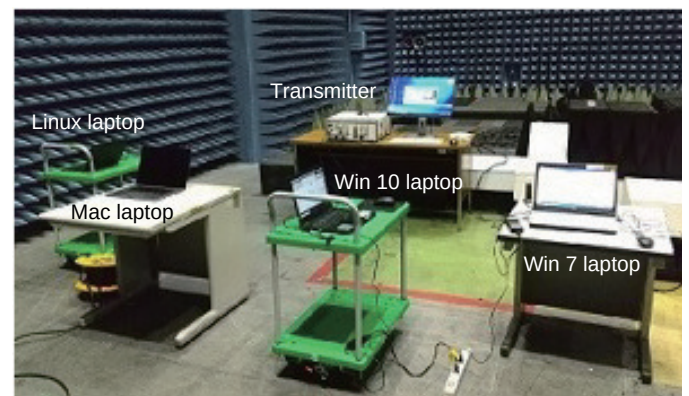


Fig. 4. Photo of experimental site.

and 4, respectively. One WLAN frame transmitter (NI PXIe-5646R) and four receiving nodes with different configurations shown in TABLE I are placed in an anechoic chamber. The WLAN frame transmitter periodically transmits IEEE 802.11g frames with a frame length of 1108 bytes at a data rate of 6 Mb/s. The four receiving nodes recode the information of received frames using a sniffing software installed into each node. Here, two nodes (“Win 7 laptop” and “Win 10 laptop”) can record only epoch time because the used WLAN device does not support TSF timestamp. On the other hand, and other two nodes (“Mac laptop” and “Linux laptop”) can record both epoch time and TSF timestamp.

Figures 5 and 6 show the empirical cumulative distribution function (CDF) of the jitter of the frame reception time at each node configuration. As shown in these figures, the jitter of TSF timestamp ranges within a few microseconds both for “Mac laptop” and “Linux laptop.” It means that TSF timestamp is quite stable index for representing the frame reception time.

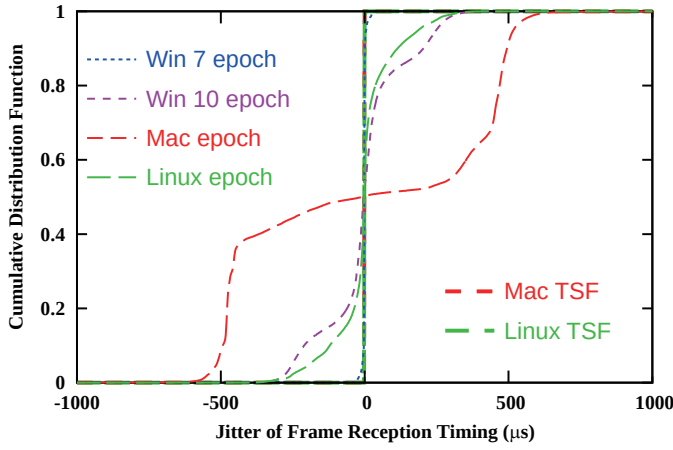


Fig. 5. CDF of jitter on frame reception time (full-range).

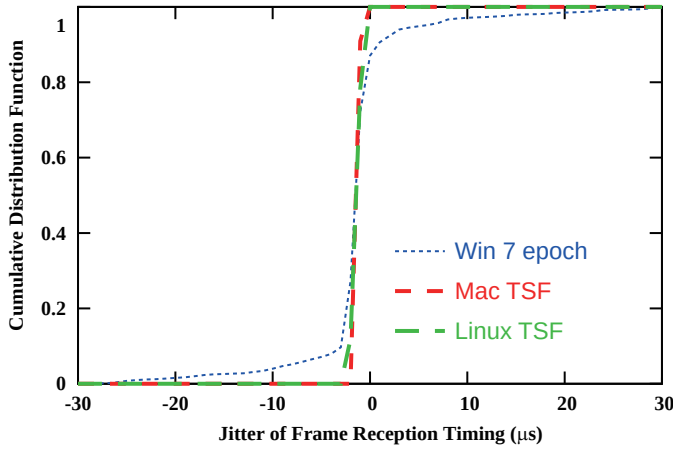


Fig. 6. CDF of jitter on frame reception time (in a small jitter range).

On the other hand, the magnitude of the jitter of epoch time is significantly larger than that of TSF timestamp except for “Win 7 laptop.” This result means that epoch time is not suitable as the timestamp of the frame reception time if μs -order time resolution is required.

IV. SIMULATION OF PROPOSED TIMESTAMP SYNCHRONIZATION METHOD

This section evaluates the performance of timestamp synchronization of the proposed method through computer simulations assuming an OBSS environment of WLAN operated in the 2.4 GHz band. The version of the WLAN is IEEE 802.11n [1], which is the latest version for the 2.4 GHz band in official. Simulation settings are shown in Table II. It is assumed that the area size is $200\text{ m} \times 200\text{ m}$, and path loss and shadowing is subjected to WINNER II B3 indoor hotspot channel [10]. There are eight BSSs in the area, and each BSS has one STA. For each STA, both downlink and uplink traffic with a offered load of 100 kBytes/s is generated.

In this simulation, it is assumed that the coordinator collects the frame reception log only from APs, and timestamp synchronization is performed among them. The clock drift

TABLE II
PARAMETERS FOR SYSTEM-LEVEL SIMULATION

Version of WLAN	IEEE 802.11n
WLAN Channel	Ch. 1 in 2.4 GHz
Signal bandwidth	20 MHz
RTS/CTS exchange	Not in use
Transmission power	20 dBm
Noise level	-94 dBm (7 dB noise figure)
Transmission data rate	6.5 Mb/s (BPSK, coding rate = 1/2)
Number of BSSs	8
Number of STAs per BSS	1
Area size	$200\text{ m} \times 200\text{ m}$
Propagation model	WINNWR II B3 indoor hotspot [10]
Direction of traffic	Downlink & uplink
Offered load per STA	100 kByte/s/link
Payload length of DATA frame	1000 Bytes
Reception log recording	AP only
Timestamp synchronization interval (T_{sync})	{30, 60, 90, 120} s
Max. clock frequency offset	25 ppm

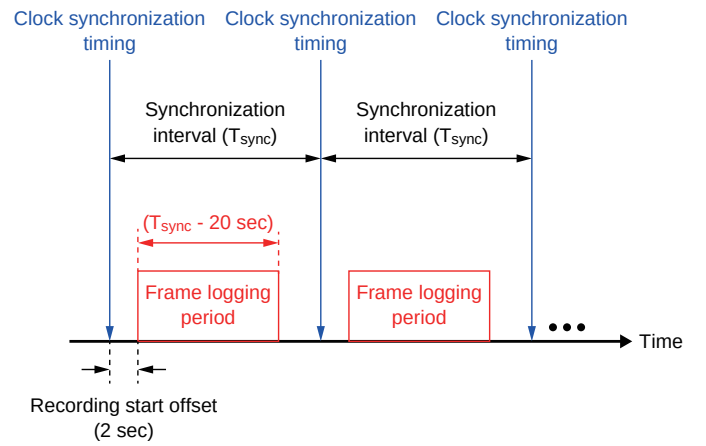


Fig. 7. Timeline of the synchronization process in the simulation.

rate is uniformly distributed within $\pm 25\text{ ppm}$; it is maximum tolerable frequency offset (clock drift rate) defined for the operation in the 2.4 GHz for IEEE 802.11n WLAN [1]. In this evaluation, the clock drift rate keeps constant in each simulation run because it was confirmed in our preliminary experiment that the clock drift rate is constant for a couple of minutes in real nodes.

The timestamp synchronization process is performed every T_{sync} seconds as shown in Fig. 7. Frame reception is logged for $(T_{\text{sync}} - 20)$ seconds, and it is assumed that the clock is perfectly synchronized among all APs by any synchronization protocol like as NTP before starting to log. The timestamp of each frame reception is assumed to fluctuate frame-by-frame according to the CDF of the timestamp jitter shown in Figs. 5 and 6. As a performance metric of the proposed timestamp synchronization method, the root mean square error (RMSE) of frame reception time is evaluated for each node configuration listed in TABLE I.

Figures 8 and 9 show the RMSE performance of the proposed timestamp synchronization method for different node

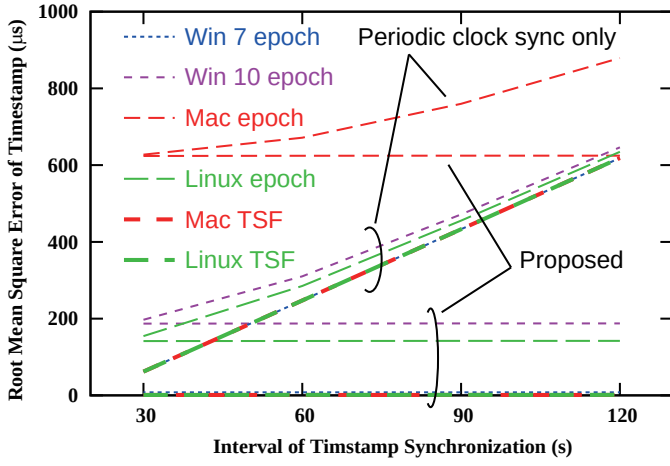


Fig. 8. RMSE of frame reception timing after clock/timestamp synchronization (full-range).

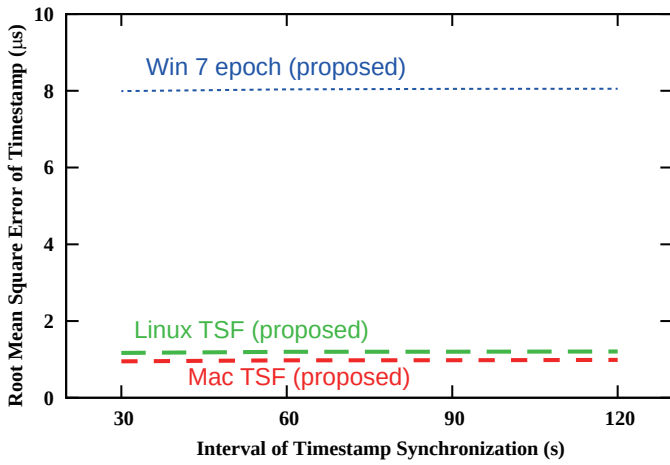


Fig. 9. RMSE of frame reception timing after clock/timestamp synchronization (in a small RMSE range).

configurations listed in TABLE I. For performance comparison, the RMSE performances in the case when only clock synchronization (i.e., without the proposed method) is performed are also shown in these graphs. From the results in these figures and Figs.5 and 6, the RMSE of timestamps increases as the synchronization interval increases if the proposed method is not employed. On the other hand, the RMSE with the proposed method is almost constant regardless with the synchronization interval, and its magnitude depends on the distribution of the jitter of original timestamp issued in each node. From these results, we can find that the coordinator can obtain the timestamps of frame reception time at each coordinating node with an error of about $1\mu s$ by using TSF timestamp and the proposed timestamp synchronization method.

V. CONCLUSION

In order to obtain fine information about frame transmission timing that is necessary to realize effective access control

in a coordinated manner among multiple nodes, this paper proposed a method to compensate for the timestamp error among multiple nodes in cooperation. In the proposed method, the coordinator collects some information about the received frames such as the received time and the information to identify each received frame. Then, the proposed method forms a node tree to determine the order of timestamp synchronization using the information of the commonly received frames among the nodes. The tree is formed so that the number of commonly received frames increases in order to improve the accuracy of the linear regression. Finally, the proposed method synchronizes the timestamps between the nodes sequentially from the root to the leaf node(s) according to the formed tree structure using the relationship of the timestamps of commonly received frames between the parent and child nodes derived by linear regression.

The jitter performance of the original timestamp obtained at each node is evaluated by experimental evaluation in an anechoic chamber using four different node configurations. This experimental evaluation revealed that TSF timestamp is obtained with a jitter of a few μs , whereas the jitter of epoch time is quite larger than that of TSF timestamp.

Finally, the system-level simulation based on IEEE 802.11n WLAN was conducted to evaluate the residual timestamp error among the coordinating nodes. The simulation results revealed that the proposed method can keep the residual timestamp error around $1\mu s$ if TSF stamp is available and the clock drift rate is stable in a couple of minutes.

ACKNOWLEDGMENT

This research and development work was supported by the MIC/SCOPE #196000002. The authors thanks to Mr. Susumu Ano for his support on the experimental evaluation.

REFERENCES

- [1] IEEE Std 802.11-2016, Dec. 2016.
- [2] Y. Seok, K. Lu, G. Bajko, J. Yee, J. Liu, and T. Pare, "Coordinated OFDMA operation," doc.: IEEE 802.11-19/1788r0, Nov. 2019.
- [3] S. Naribole, W. B. Lee, S. Kandala, R. Duan, and A. Ranganath, "Shared TXOP protocol," doc.: IEEE 802.11-20/0277r0, March 2020.
- [4] Y. Zhu, Z. Niu, Q. Zhang, B. Tan, Z. Zhou, and J. Zhu, "A multi-AP architecture for high-density WLANs: protocol design and experimental evaluation," in Proc. IEEE SECON, pp. 28–36, June 2008. DOI:10.1109/SAHCN.2008.14
- [5] T. Nakahira, K. Ishihara, Y. Asai, Y. Takatori, R. Kudo, and M. Mizoguchi, "Centralized control of carrier sense threshold and channel bandwidth in high-density WLANs," in Proc. IEEE APMC 2014, pp. 570–572, Nov. 2014.
- [6] T. H. Lim, W. S. Jeon, and D. G. Jeong, "Centralized channel allocation scheme in densely deployed 802.11 wireless LANs," in Proc. IEEE ICACT 2016, pp. 249–253, Jan. 2016. DOI:10.1109/ICACTION.2016.7423347
- [7] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep reinforcement learning-based channel allocation for wireless LANs with graph convolutional networks," IEEE Access, vol. 8, no. 1, pp. 31823–31834, Feb. 2020. DOI:10.1109/ACCESS.2020.2973140
- [8] Y. Kihira, Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, "Adversarial reinforcement learning-based robust access point coordination against uncoordinated interference," in Proc. IEEE VTC2020-Fall, Nov.–Dec. 2020.

- [9] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in Proc. Intl. Conf. on Machine Learning (ICML), pp. 2817–2826, Aug. 2017.
- [10] "WINNER II channel models, Part I Channel models" IST-4-027756 WINNER II, D1.1.2 V1.2, Feb. 2008.

Kazuto Yano received the B.E. degree in electrical and electronic engineering, and the M.S. and Ph.D. degrees in communications and computer engineering from Kyoto University in 2000, 2002, and 2005, respectively. He was a research fellow at the Japan Society for the Promotion of Science (JSPS) from 2004 to 2006. In 2006, he joined the Advanced Telecommunications Research Institute International (ATR). Currently, he is the Head of Dept. Wireless Communication Systems at Wave Engineering Laboratories, ATR. His research interests include spacetime signal processing for interference suppression, MIMO transmission, and PHY/MAC cross-layer design of wireless communication systems for ISM bands. He received the IEEE VTS Japan 2001 Researcher's Encouragement Award, the IEICE Young Researcher's Award in 2007, the Ericsson Young Scientist Award 2007 and the IEICE 2007 Active Research Award in Radio Communication Systems. He also received 2010 Young Investigator Award in Software Radio from IEICE Technical Committee on Software Radio. In addition, he received IEICE Communications Society Best Tutorial Paper Award in 2017, and ICAIIC 2019 Excellent Paper Award in 2019. He is a member of IEEE, and a senior member of IEICE.

Eiji Nii received his B.S., M.S., and Ph.D. degrees from Kansai University, Osaka, Japan, in 2015, 2017, 2020, respectively. He is currently a researcher at Wave Engineering Laboratories of Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan in 2020. His research interest is in control multiple autonomous mobile devices with swarm intelligence. He is a member of IPSJ and IEICE.

Kenta Suzuki received his B.S. degree in Computer Science from Hirosaki university, Japan in 2011, and received his M.S. degree in Computer Science from Tohoku university, Sendai, in 2013. In 2013, he joined Mobile Techno Corp., where he was engaged in development on broadcast radio system for disaster prevention and system level simulator for network simulation. Since April 2019, he has been assigned to Advanced Telecommunications Research Institute International (ATR) as a researcher, and he is engaged in research and development on network scan systems. He is a member of the IEICE.

Koji Yamamoto received the B.E. degree in electrical and electronic engineering from Kyoto University in 2002, and the master and Ph.D. degrees in Informatics from Kyoto University in 2004 and 2005, respectively. From 2004 to 2005, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). Since 2005, he has been with the Graduate School of Informatics, Kyoto University, where he is currently an associate professor. From 2008 to 2009, he was a visiting researcher at Wireless@KTH, Royal Institute of Technology (KTH) in Sweden. He serves as an editor of IEEE Open Journal of Vehicular Technology, IEEE Wireless Communications Letters, and Journal of Communications and Information Networks, a track co-chair of APCC 2017, CCNC 2018, APCC 2018, and CCNC 2019, and a vice co-chair of IEEE ComSoc APB CCC. He was a tutorial lecturer in IEEE ICC 2019. His research interests include radio resource management, game theory, and machine learning. He received the PIMRC 2004 Best Student Paper Award in 2004, the Ericsson Young Scientist Award in 2006. He also received the Young Researcher's Award, the Paper Award, SUEMATSU-Yasuharu Award, Educational Service Award from the IEICE of Japan in 2008, 2011, 2016, and 2020, respectively, and IEEE Kansai Section GOLD Award in 2012. He is a senior member of the IEEE and a member of the Operations Research Society of Japan.